S-Learning: A Biomimetic Algorithm for Learning, Memory, and Control in Robots

Brandon Rohrer

***Abstract*— S-Learning is a sequence-based learning algorithm patterned on human motor behavior. Discrete-time and quantized sensory information is amassed in real-time to form a dynamic model of the system being controlled and its environment. No explicit model is provided *a priori*, nor any hint about what the structure of the model might be. As the core of a Brain-Emulating Cognition and Control Architecture (BECCA), S-Learning provides a mechanism for human-inspired learning, memory, and control in machines. In a simulation of a point-to-point reaching task, S-Learning demonstrates several attributes of human motor behavior, including learning through exploration and task transfer.**

I. INTRODUCTION

The field of "Learning to learn," also termed *generalization* or *bias learning*, takes machine performance a step further than many learning algorithms. [1] Generalization algorithms seek to improve system performance not just on tasks for which the systems explicitly train, but also on novel, unrelated tasks. Humans are often able to learn a task after only one or two exposures due to the ability to generalize from previously learned tasks. Generalization algorithms attempt to imbue automated systems with this same ability. Common approaches include connectionist networks [2], [3], statistical (including Bayesian, memory-based, and Markovian) methods [4], [5], [6], dimensionality reduction [7], and modified reinforcement learning techniques [8], [9]. A subset of generalization algorithms are explicitly biologically-motivated. They mimic the human brain, as it serves as an existence proof for solutions to daunting perception and control problems. S-Learning falls into this category.

A. Relation to Temporal-Difference techniques

S-learning is a variant of temporal-difference (TD) learning. It is superficially similar to Q-learning, another TD algorithm, but involves *sequences* of discrete events (hence the *S*). TD algorithms are typically effective at discovering optimal sequences of actions in unknown environments. However, existing algorithms almost exclusively address the static TD problem, in which the states that result in reward or punishment are fixed. This is equivalent to a control system that has a fixed goal that does not vary over time. And while multiple instances of a static TD algorithm, such as Q-learning, can be employed to account for multiple goal states, the experience gained while training one does not transfer to others in a straightforward way. Such an approach typically requires a separate training period for each instance of the algorithm. Even when this multiple-instance approach is successful, it still does not aid the system in reaching unfamiliar goal states.

The distinguishing characteristic of S-learning is that it continually records recurring patterns to build a library of past experiences. This library allows a goal-seeking agent to piece the patterns together to form a complete path to a goal. The strength of this approach is that the goal can be any previously-visited state, not just one or a few that were hard-coded from the start. Thus S-learning provides a potential solution to the dynamic TD problem.

B. Rohrer is a member of the Intelligent Systems, Robotics, and Cybernetics Group at Sandia National Laboratories, Albuquerque, NM, USA. Email: `brrohre@sandia.gov`

B. Relation to Markov Models

A set of sequences of length two is similar to a Markov model. The likelihood of transitioning from state A to state B can be inferred from the sequence set and could alternatively be represented in matrix form. Similarly, longer sequences could be represented as higher-order Markov models. It is accurate to describe an S-Learning sequence library as a shorthand way of representing a series of Markov models of order one to order $N-1$, where N is the maximum sequence length. The advantage of a sequence library is that it is concise. A first order Markov model in a system with M possible states can be represented by a $M \times M$ matrix, a second order Markov model by a $M^2 \times M$ matrix, and an $N-1$ order Markov model by a $M^{N-1} \times M$ matrix. For the system simulated in this paper, in which $N = 5$ and $M = 2^{234}$, this representation quickly becomes computationally burdensome. In this sense, a sequence library is a sparse matrix coding for a multi-order Markov model.

II. METHOD

A. Architecture

S-Learning is at the core of a biomimetic Brain-Emulating Cognition and Control Architecture (BECCA, see Fig. 1). BECCA consists of an Agent, a Planner, a World, and an S-Learning Engine, each of which is briefly described below.

Agent. The Agent sets goals for the system. The goals are expressed in terms of the sensory state information available from the World. Goals can be a specific state, a set of states, or a portion of a state. Multiple, even conflicting, goals can exist. Goals can change over time, and the Agent can use new state information to decide when and how to change them. The current set of goals is available for use by the Planner.

Planner. The Planner determines which (if any) actions to take at any given point in time. It takes in goals from the Agent and current state information to inform its decisions.

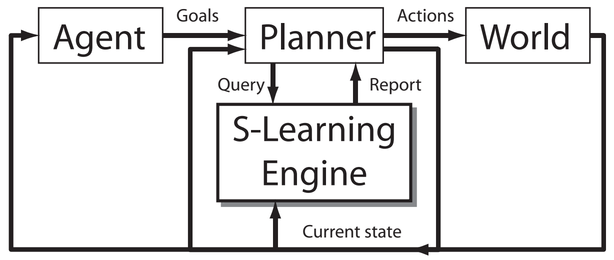

Fig. 1. Brain-Emulating Cognition and Control Architecture (BECCA), featuring S-Learning: a block diagram representation. The S-Learning algorithm is used as an engine to bootstrap a model of the World. This model is referenced by the Planner and uses new state information to refine its world model in order to achieve goals provided by the Agent.

The Planner queries the S-Learning Engine in order to predict the results of possible courses of action. Exploratory actions are also considered, particularly if the current state is unfamiliar and the S-Learning Engine cannot predict a path to a goal state. After a course of action is determined, the Planner issues commands to the World and reports those actions in the state vector.

World. The World is the external system that is being learned and controlled. It is analogous to the Plant in classical control system formulations. The World can either be simulated or instantiated in hardware, but in either case, the only information it provides back to the rest of BECCA is through its sensors. This means that in simulations, BECCA does not have access to the World's internal variables.

S-Learning Engine. The S-Learning Engine uses the regularly-updated stream of state information to bootstrap a model of the World. There is no explicit model, assumed dynamics, or implied structure. Instead, the S-Learning Engine observes repeated state sequences, particularly those that result in a goal state. These state sequences are stored in a library, which is referenced by the Planner during action planning. The S-Learning Engine also keeps track of the sequences that the Planner selects as action plans. If a sequence leads to a goal, as predicted, it is reinforced and weighted more heavily in the sequence library. If a sequence fails to lead to a predicted goal, its weighting in the library is reduced. After a number of failed predictions, a sequence becomes sufficiently weak that it is dropped from the library.

B. *S-Learning Algorithm*

S-Learning provides a single mechanism for handling learning, memory, and prediction in BECCA. The learning and memory behavior of S-Learning emerge from the way new states are incorporated into the bootstrapped world model. Initially, the sequence library, κ, has no prior experiences and contains no state sequences. When a goal state is achieved (presumably though the exploratory efforts of the Planner) the sequence of events leading up to the goal are stored in the library.

Prediction in S-Learning is straightforward:

1) Begin with the most recently observed state ν_0.
2) Find the set of all the sequences, χ, in the sequence library, κ, that begin with ν_0. Call this set χ_π.
3) For each χ in χ_π find any goal states in the sequence. Those containing goal states from, $\hat{\chi}_\pi$, a subset of χ_π. $\hat{\chi}_\pi$ comprises possible plans of action for reaching a goal state.

When the S-Learning Engine is queried by the Planner, it reports back with $\hat{\chi}_\pi$, which the Planner includes in its selection process. Other, more sophisticated prediction methods based on the sequence library are possible as well. For instance, daisy-chaining sequences together, creating trees of possible plans, would allow the Planner to create novel plans and generate a series of sub-goals.

If the Planner utilizes a sequence from the library to form a prediction and a plan of execution, then it has an expectation about when a goal will be achieved. If a goal is not achieved when expected, that sequence of events is appended to the library, allowing future prediction of the same failure.

If a goal is achieved when expected, then the successful sequence is compared against the sequence used in prediction, and their common elements are used to form a new sequence. This *intersection sequence* contains information that is relevant to achieving the goal, but tends to omit sources of noise and irrelevant sensory information. To use an everyday example, humans do not factor in the color of a cup when planning to grasp it; in this task, color is irrelevant. Likewise, intersection sequences contain only the information that is consistently present during successful sequences, and leaves out information that is not.

C. *Mimicking biology*

The structure of the state information and dynamics by which it is modified are based loosely on neural processes. The state that is passed from the World to the S-Learning Engine is a vector of ones and zeros, representing an array of neurons that are either firing or at rest. Each state is represented in the sequence library as an array of binary numbers, representing synaptic connections between sensory neurons and a central neuron or neural circuit that represents that state. The sequence representation within the library is an acknowledgement of the brain's propensity to store events sequentially, as mediated by the hippocampus [10] and instantiated in thalamocortical circuits. [11], [12] Both unexpected achievement of a goal state and unexpected failure to achieve a goal state are followed by alteration of synaptic weights, one model of neuroplasticity. Even the discrete nature of sensory states and actions within BECCA and S-Learning is motivated by a growing body of evidence that human sensory [13], [14], [15], [16] and motor [17], [18], [19], [20], [21], [22] processes are inherently discrete, modulated by rhythmic gamma and theta activity in the brain. [23], [24], [25], [26], [11]

D. *Simulation*

S-Learning was applied to a simulation of a human point-to-point reaching task. The World, consisting of an anatomically approximate model of a human head, torso, and arm, including inertial properties, was simulated in MATLAB. (Fig. 2) Control inputs to the arm from the Planner comprised

TABLE I
COMPOSITION OF THE STATE VECTOR IN THE REACHING SIMULATION.

Sensory modality or command type	Number of state elements
command: shoulder flexion	4
command: shoulder extension	4
command: elbow flexion	4
command: elbow extension	4
vision: target x-position	10
vision: target y-position	10
vision: target radial error	10
vision: target angular error	8
vision: target angular error fine	36
position: coarse shoulder	11
position: coarse elbow	13
position: fine shoulder	55
position: fine elbow	65
velocity: shoulder	6
velocity: elbow	6
Total:	246

a set of position steps in joint equilibrium position with minimum-jerk transitions. The sensory information passed from the World back to BECCA included angular velocity, coarse and fine angle information for both the shoulder and elbow joints, coarse vision of the position of the target, and coarse vision of the position error between the target and the hand. Each of these sensory modalities was binned (quantized) and expressed in terms of a vector of binary states. (Table I.)

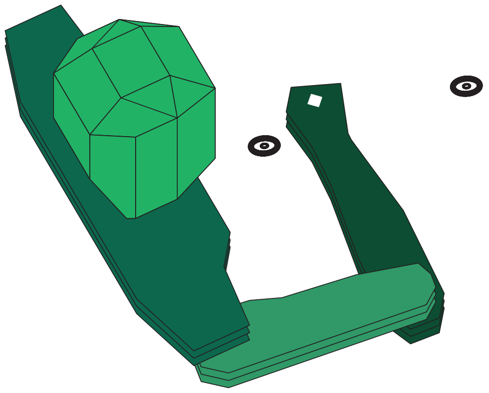
Fig. 2. A graphical representation of the simulated human performing a reaching movement.

The simulated reaching task consisted of two alternating subtasks. Interspersing them allowed evaluation of S-Learning's ability to learn, to generalize that learning to a new task, and to retain that learning in the face of potential interference.

1) Subtask A: The first simulated task consisted of moving the hand to a target position on the "subject's" midline, 15 cm from the sternum. Once the center of the hand arrived within 4 cm of this position, a new target was presented on the midline, 45 cm from the sternum. After the center of the hand came within 4 cm of this target, the cycle started over again.

2) Subtask B: The second task was identical to the first, with the exception that the targets were both shifted 10 cm to the right.

As currently implemented in MATLAB, BECCA is structured as an iterative loop. For the first 10,000 iterations of the BECCA loop, Subtask A was presented. This was followed by 5,000 iterations of Subtask B, and then completed with another 5,000 iterations of Subtask A.

III. RESULTS

Early attempts to reach the goals consisted primarily of exploratory wandering (random movements). After a few chance successful sequences were added to the library, these served as a basis for better-directed movements, which resulted in more successful sequences. This process produced progressively more direct movement sequences, resulting in goals being reached more rapidly and more frequently. (Fig. 3) The number of goals reached in each task per 1000-iteration block are shown in Fig. 4.

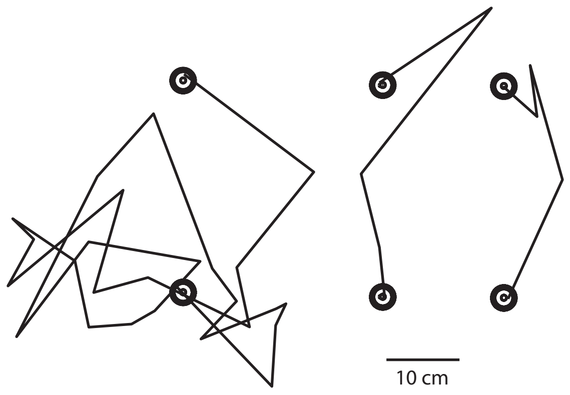

Fig. 3. Typical paths to goal at various stages of learning. Initial movements resemble "motor babbling" observed in infants and hypothesized as a primary mechanism for motor learning. Later movements become more efficient, resulting in shorter paths, smoother movements, fewer submovements, lower actuation torques, and more rapid goal achievement.

IV. DISCUSSION

The most striking aspect of Fig. 4 is that time spent learning one task appears to transfer to the other. The learning curve in Epoch B is much steeper than in A1, presumably because some of the experience gained in A1 generalized to B. Similarly, the experience gained during Epoch B appears to have generalized to A2; the initial performance levels in A2 are higher than those at the conclusion of A1. The two tasks are distinctly different as far as the algorithm is concerned; there is no explicit representation that allows S-Learning to make this transfer. Transfer between similar, complementary tasks is a feature of human motor learning as well. Contrast this with two common approaches to this type of learning task, Q-Learning and connectionist networks, which typically require either a large amount of re-training time when a task is changed or multiple instances of the respective algorithms under a supervisory context-switching module. S-Learning mimics human performance in that only

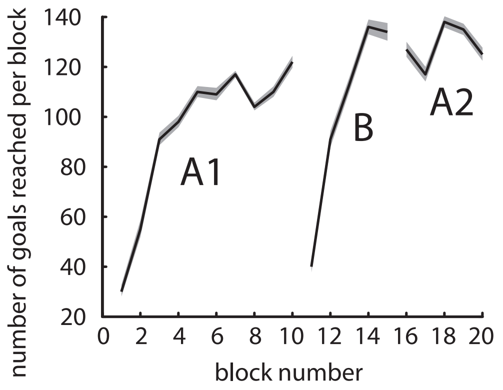

Fig. 4. Goals reached during 1000-iteration blocks. Grey bands show plus and minus one standard deviation. Typically, one movement was executed per iteration of the BECCA loop. During epoch A1, performance increased quickly during the first 6 blocks, slowing after that. Epoch B saw an initial drop in performance as the S-Learning algorithm began learning the new task, but performance increased more rapidly and to a higher level than in A1. Epoch A2 showed no great drop in performance due to after effects of B, but rather started out at a higher level of performance than it showed when A1 terminated.

a single, unmodified instance was needed to learn both tasks shown here.

S-Learning is an extremely general learning approach. The same S-Learning and BECCA implementation shown here could have been applied to stabilizing an inverted pendulum, learning to grasp with a robotic hand, or steering an unmanned vehicle. As long as the state contains appropriate sensor information and the system has adequate actuation, S-Learning can be used to learn its behavior, store it in memory, and recall it for use in control. Many other methods have been used to control two-link robots; it is not a difficult control problem. The significance of S-Learning accomplishing the task is that it did so without knowing *a priori* how to interpret any of its sensor data or how to reach its goal.

V. Conclusion

By basing its function on observed psychophsyiology, S-Learning is able to recreate some of the salient features and strengths of human motor behavior. This paper has demonstrated a simple simulation of S-learning in point-to-point reaching. However, the general nature of the algorithm suggests that it may also be capable of solving more complex motor control problems, including grasp, bimanual manipulation, visual tracking, balance, and bipedal locomotion. Future simulations and hardware implementations of S-Learning will test whether this is the case.

VI. Acknowledgments

The generous technical and programmatic support of Kristopher Klingler, Fred Rothganger, Larry Shipers, and Patrick Xavier are gratefully acknowledged. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

References

[1] S. Thrun and L. Pratt, *Learning to Learn*. Kluwer Academic Publishers, 1998, ch. Learning to Learn: Introduction and Overview, pp. 3–18.

[2] L. Pratt and B. Jennings, "A survey of connectionist network reuse through transfer," *Connection Science*, vol. 8, no. 2, 1996.

[3] R. Caruana, *Learning to Learn*. Kluwer Academic Publishers, 1998, ch. Multitask learning, pp. 95–134.

[4] J. Baxter, *Learning to Learn*. Kluwer Academic Publishers, 1998, ch. Theoretical models of learning to learn, pp. 71–94.

[5] D. Shepard, "A two-dimensional interpolation function for irregularly spaced data," in *23rd National Conference ACM*, 1968.

[6] J. Tenenbaum, *Advances in Neural Information Processing Systems 11*. Cambridge, MA: MIT Press, 1999, ch. Bayesian modeling of human concept learning.

[7] N. Intrator and S. Edelman, "Making a low-dimensional representation suitable for diverse tasks," *Connection Science*, vol. 8, no. 2, 1996.

[8] J. Schmidhuber, J. Zhao, and N. Schraudolph, *Learning to Learn*. Kluwer Academic Publishers, 1998, ch. Reinforcement learning with self-modifying policies, pp. 293–310.

[9] R. Maclin and J. Shavlik, *Learning to Learn*. Kluwer Academic Publishers, 1998, ch. Creating advice-taking reinforcement learners, pp. 311–347.

[10] W. Levy, A. Sanyal, P. Rodriguez, D. Sullivan, and X. Wu, "The formation of neural codes in the hippocampus: trace conditioning as a prototypical paradigm for studying the random recording hypothesis," *Biological Cybernetics*, vol. 92, pp. 409–426, 2005.

[11] A. Rodriguez, J. Whitson, and R. Granger, "Derivation and analysis of basic computational operations of thalamocortical circuits," *J Cog Neuroscience*, vol. 16, no. 5, pp. 856–877, 2004.

[12] R. Granger, S. Petrovic, A. Felch, J. Kerr, M. Johnson, C. Wuerth, and J. Benvenuto, "Engines of the brain: the computational "instruction set" of perception and cognition," 2005. [Online]. Available: http://www.ics.uci.edu/ granger/enginesRHG.pdf

[13] W. James, *The Principles of Psychology*. New York: Dover Publications, 1890.

[14] D. Purves, J. A. Paydarfar, and T. J. Andrews, "The wagon wheel illusion in movies and reality," *Proceedings of the National Academy of Sciences USA*, vol. 93, pp. 3693–3697, 1996.

[15] M. Gho and F. J. Varela, "A quantitative assessment of the dependency of the visual temporal frame upon the cortical rhythm," *J Physiol. Paris*, vol. 83, pp. 95–101, 1988.

[16] R. VanRullen and C. Koch, "Is perception discrete or continuous?" *Trends in Cognitive Sciences*, vol. 7, pp. 207–213, 2003.

[17] A. B. Vallbo and J. Wessberg, "Organization of motor output in slow finger movements in man," *Journal of Physiology*, vol. 469, pp. 673–691, 1993.

[18] R. S. Woodworth, "The accuracy of voluntary movement," *Psychology Review Monogr Suppl*, 1899.

[19] C. von Hofsten, "Structuring of early reaching movements: A longitudinal study," *Journal of Motor Behavior*, vol. 23, no. 4, pp. 280–292, 1991.

[20] H. I. Krebs, M. L. Aisen, B. T. Volpe, and N. Hogan, "Quantization of continuous arm movements in humans with brain injury," *Proceedings of the National Academy of Science*, vol. 96, pp. 4645–9, April 1999, neurobiology.

[21] B. Rohrer, S. Fasoli, H. Krebs, B. Volpe, W. Frontera, J. Stein, and N. Hogan, "Submovements grow larger, fewer, and more blended during stroke recovery," *Motor Control*, vol. 8, no. 4, pp. 472–483, 2004.

[22] T. E. Milner, "A model for the generation of movements requiring endpoint precision," *Neuroscience*, vol. 49, pp. 365–374, 1992.

[23] J. Wessberg and N. Kakuda, "Single motor unit activity in relation to pulsatile motor output in human finger movements," *The Journal of Physiology*, vol. 517, pp. 273–285, 1999.

[24] J. McAuley, J. Rothwell, and C. Marsden, "Human anticipatory eye movements may reflect rhythmic central nervous activity," *Neuroscience*, vol. 94, no. 2, pp. 339–350, 1999.

[25] J. Gross, L. Timmermann, J. Kujala, M. Dirks, F. Schmitz, R. Salmelin, and A. Schnitzler, "The neural basis of intermittent motor control in humans," *Proceedings of the National Academy of Science*, vol. 99, no. 4, pp. 2299–2302, February 2002.

[26] W. Singer, "Neuronal synchrony: a versatile code for the definition of relations?" *Neuron*, vol. 24, pp. 49–65, 1999.